Ares I-X Ground Diagnostic Prototype

Mark Schwabacher and Rodney Martin
NASA Ames Research Center, Moffett Field, CA

Robert Waterman
NASA John F. Kennedy Space Center, Cape Canaveral, FL

Rebecca Oostdyk
ASRC Aerospace Corporation, Cape Canaveral, FL

John Ossenfort and Bryan Matthews
Stinger Ghaffarian Technologies, Inc., Moffett Field, CA

April 27, 2010

Introduction

- Automated pre-launch diagnostics for launch vehicles offers the following potential benefits:
  - Improved safety
  - Reduced cost
  - Reduced launch delays
- Can include data from vehicle assembly, and from assembled vehicle while it is on the launch pad (from pre-launch umbilical)

What is Ares I-X?

Ares I-X was the first test flight of Ares I. It had a dummy second stage and a dummy capsule, and launched on 10/28/2009.

Definitions

- **Anomaly detection:** detecting that something is unusual
- **Failure**: The unacceptable performance of intended **function**.
- **Failure Detection:** Deciding that a **failure** exists.
- **Fault Diagnosis**: Determining the possible causes of a **failure**.
- **Fault Isolation**: Determining the possible locations of a hypothesized **failure** cause, within a defined level of granularity.
- All are part of Integrated Systems Health Management (ISHM)
- All take as input sensor values and command stream.

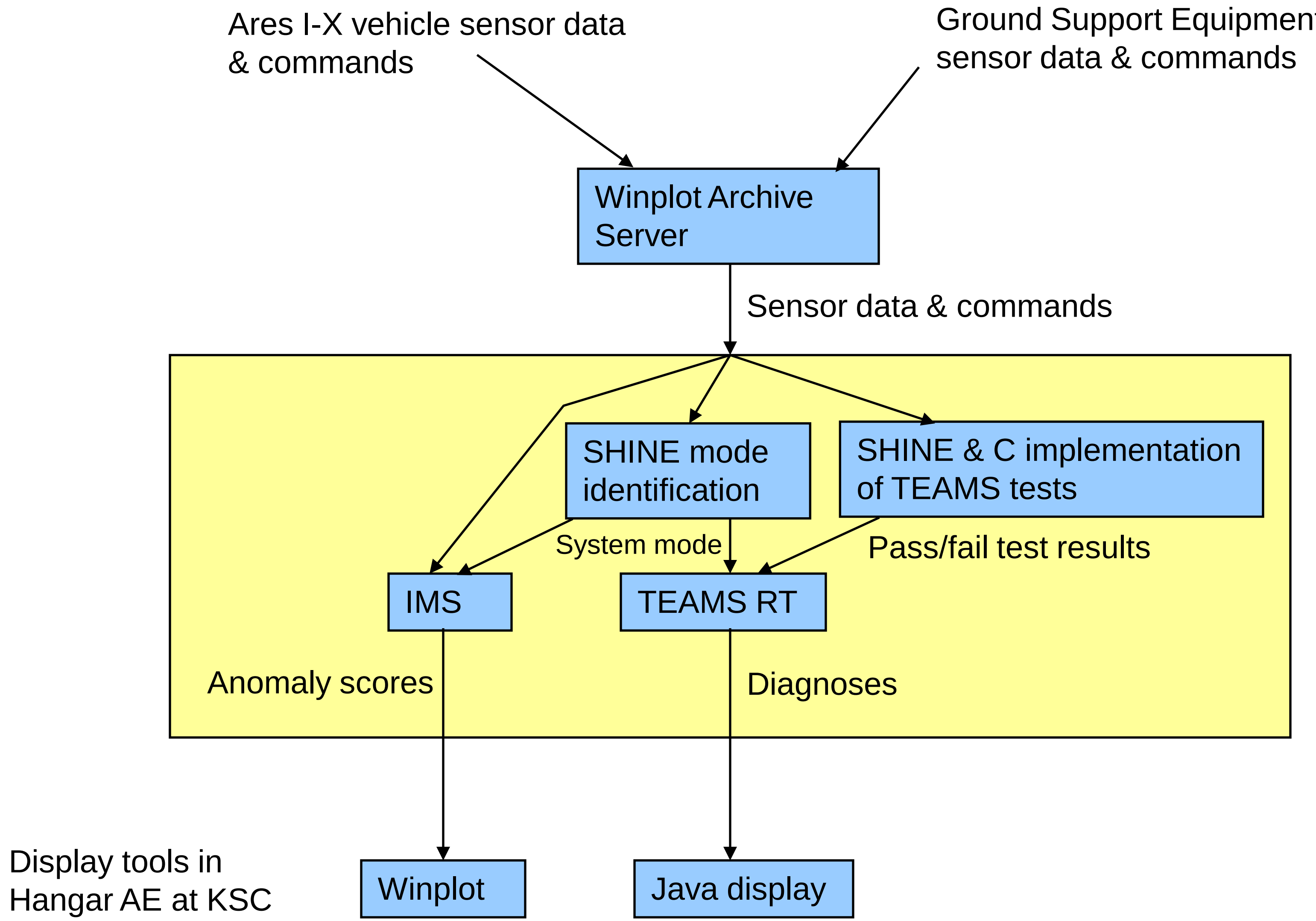
Scope of Ares I-X Ground Diagnostic Prototype

- Anomaly detection, failure detection, fault isolation, and fault diagnosis for Ares I-X while it is in the Vehicle Assembly Building and while it is on the launch pad
- Focused on the first-stage thrust vector control (TVC) and the associated ground hydraulics
- Deployed diagnostic software to Hangar AE at NASA KSC
- Used near-real-time data from the VAB and the pad
- Integrated the diagnostic software with existing software at NASA KSC
- Assessed the difficulty of certifying the software for human spaceflight
- Intended to serve as a prototype of the ground diagnostic system for Ares I

Tools used in GDP

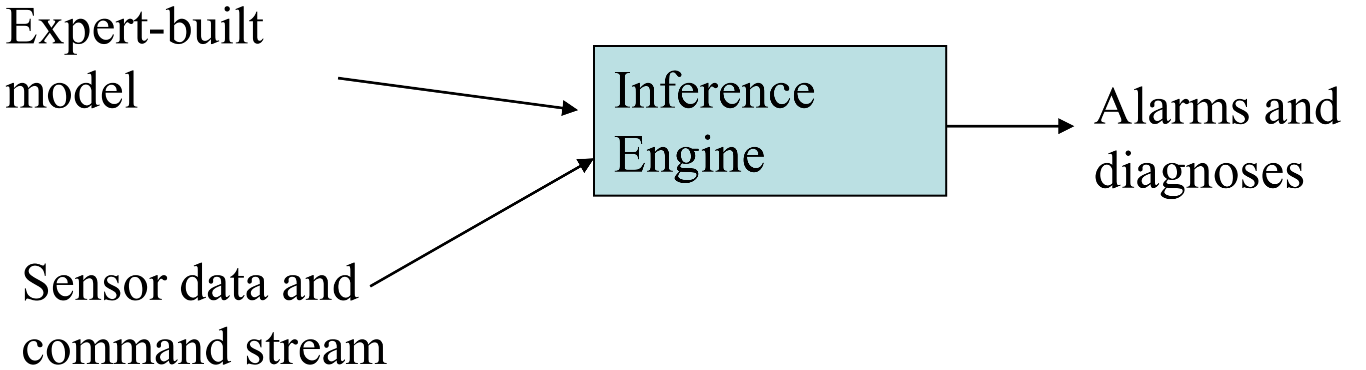
- TEAMS-RT, a model-based tool for fault isolation and diagnosis
- IMS, a data-driven tool for anomaly detection
- SHINE, a rule-based tool that we are using for failure detection

Simplified GDP Architecture



Model-based Diagnostics

- Human experts build a hierarchical model of the system.
- The model describes how the components of the system should work, their interconnections, and their failure modes.
- The component models can be physics-based or finite state machines.
- An inference engine uses the model and the real-time data and command stream to determine the state of the system and to diagnose failures, including reasoning about multi-component failures.
- Ames' Livingstone and HyDE, JPL's MEXEC, and TEAMS RT (commercial product) use this approach.

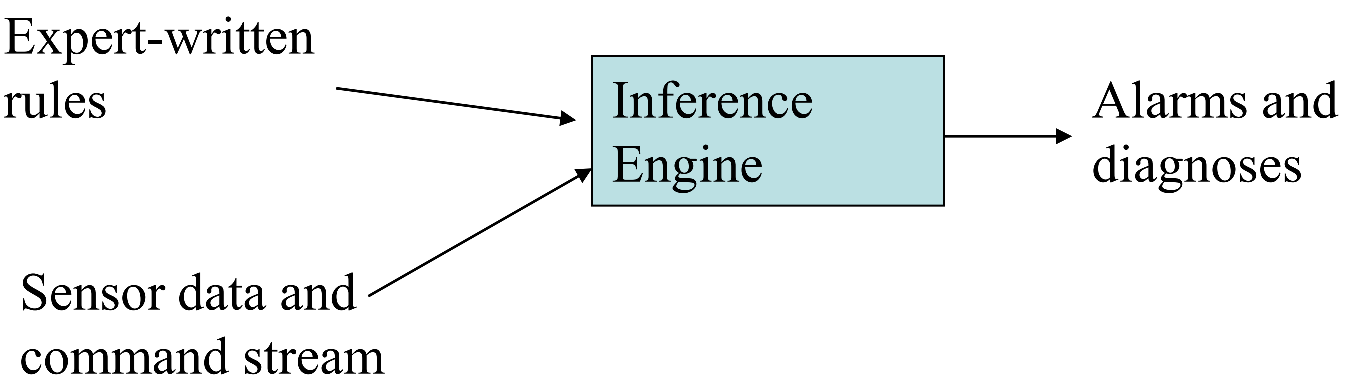


TEAMS RT

- Model-based tool for fault isolation and diagnosis
- Commercial product from QSI; developed using NASA ARC SBIR funding
- Uses a qualitative model of failure propagation
- Is being used by Honeywell to model the Orion CEV under subcontract to Lockheed Martin (will be certified and flown)
- Requires “wrapper” code to convert sensor values into “pass/fail” test results and to identify the system mode.

Rule-based expert systems

- Human experts write rules in a special-purpose language.
- The rules map fault signatures to faults.
- An inference engine decides which rules are applicable and executes them.
- JPL's SHINE uses this approach (TRL 9 heritage)
- G2 (commercial product) also uses this approach.

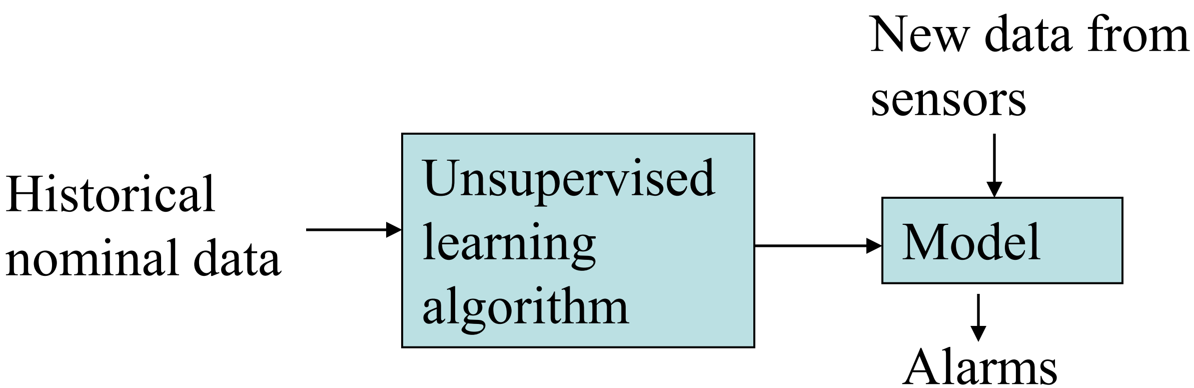


SHINE

- Rule-based expert system
- Uses a data-flow representation to execute rules efficiently
- Basis for four of the nine BEAM components
- Very fast: Over 300 million rules per second on current desktop computers
- Extremely small memory and storage footprint, suitable for embedded applications
- Applied to several mission ground ops including Voyager and EUVE
- Tested on flight hardware (X-33 AFE) and in flight (DFRC F/A-18)
- We used SHINE to provide the “wrappers” for TEAMS-RT
  - Mode and event identification
  - “pass/fail” test results

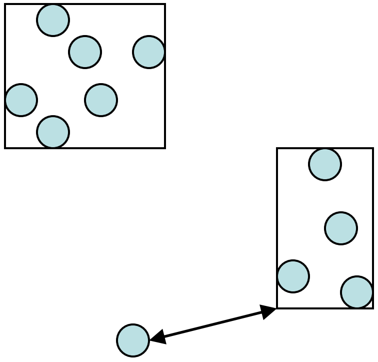
Unsupervised Anomaly Detection

- Unsupervised anomaly detections algorithms look for portions of the data that are different from the rest of the data (outliers).
- Unsupervised approach to fault detection uses only nominal training data, and learns a model of the nominal data. When new data doesn’t match the model, it signals an alarm.
- Can catch unknown faults
- Does fault detection only, not diagnosis
- Useful when few examples of faults are available
- Can detect interactions among hundreds of variables
- Ames’ IMS and Orca and JPL’s BEAM use this approach. (BEAM can also use a physics-based model and/or supervised learning)



IMS

- Data-driven unsupervised learning algorithm
- Inductive Monitoring System (IMS)
  - clusters the training data
  - uses distance to nearest cluster as anomaly measure
- Developed by Dave Iverson of ARC
- Is currently running on a console at JSC MCC to find anomalies in live ISS CMG data and has been certified as Class C software for that application.
- Also used to find anomalies in historical SSME data
- Runner-Up in the 2008 NASA Software of the Year Competition
- Generic C++ code



Data used for testing our software

- Before the Ares I-X data became available, we used historical Shuttle TVC and HSS data to train and test our software.
- We inserted simulated failures into the historical Shuttle data.
- The prototype ran on live Ares I-X data from the VAB and from the pad.

Simulated failure modes in Shuttle data

- FSM (Fuel Supply Module) pressure drop due to N_2H_4 (Hydrazine) leak
- Hydraulic pumping unit over-temperature failure
- Hydraulic fluid reservoir level drop due to hydraulic fluid leak
- Actuator stuck during actuator positioning test

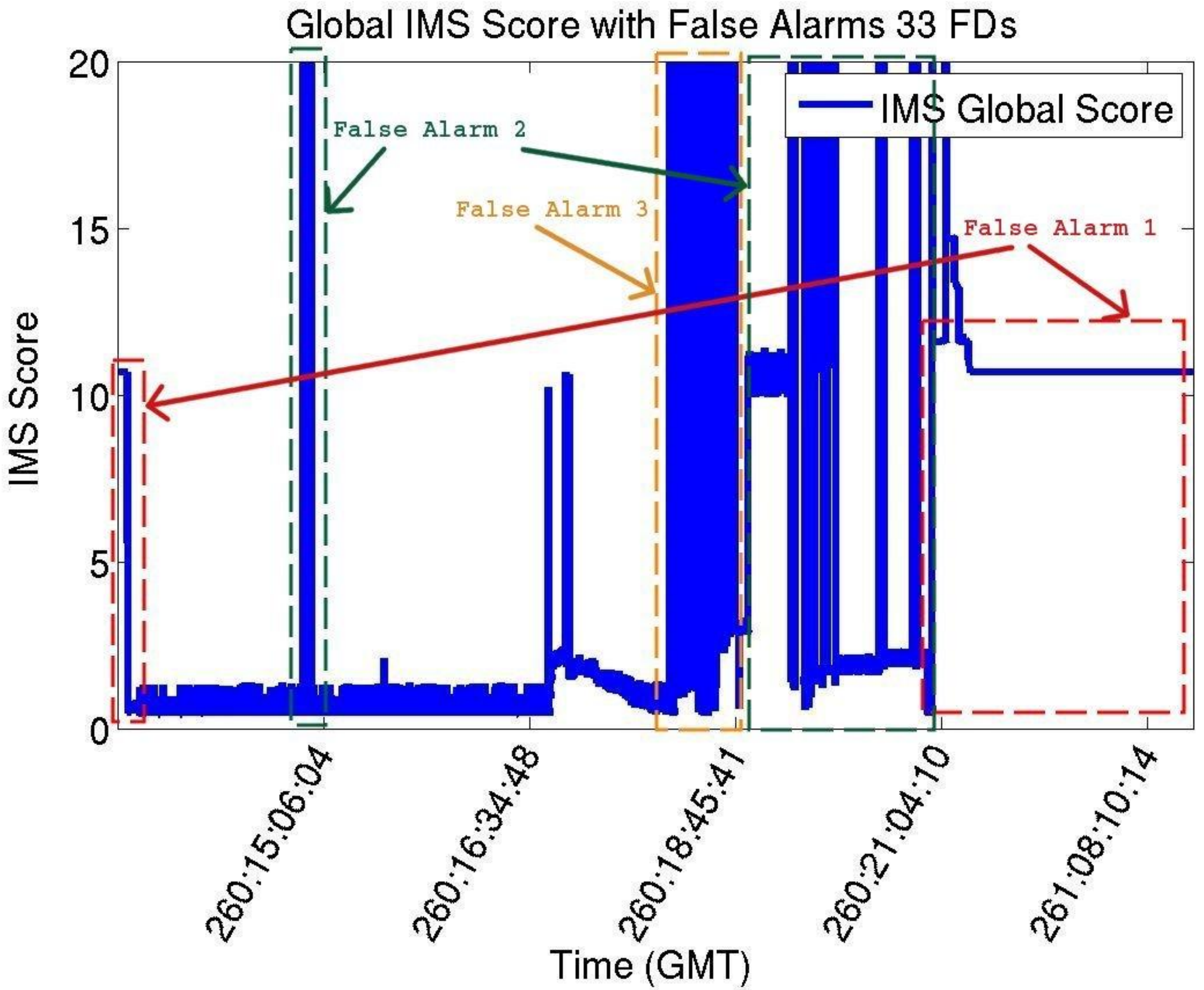
TEAMS/SHINE results on Shuttle data

- Tested the prototype on data from 7 Shuttle flights with simulated failures
- Testing revealed some bugs, which we fixed
  - Some bugs caused by incorrect assumptions about TVC testing procedures
- After fixing bugs, prototype ran on all 7 flights with all simulated failures correctly detected and no false alarms.

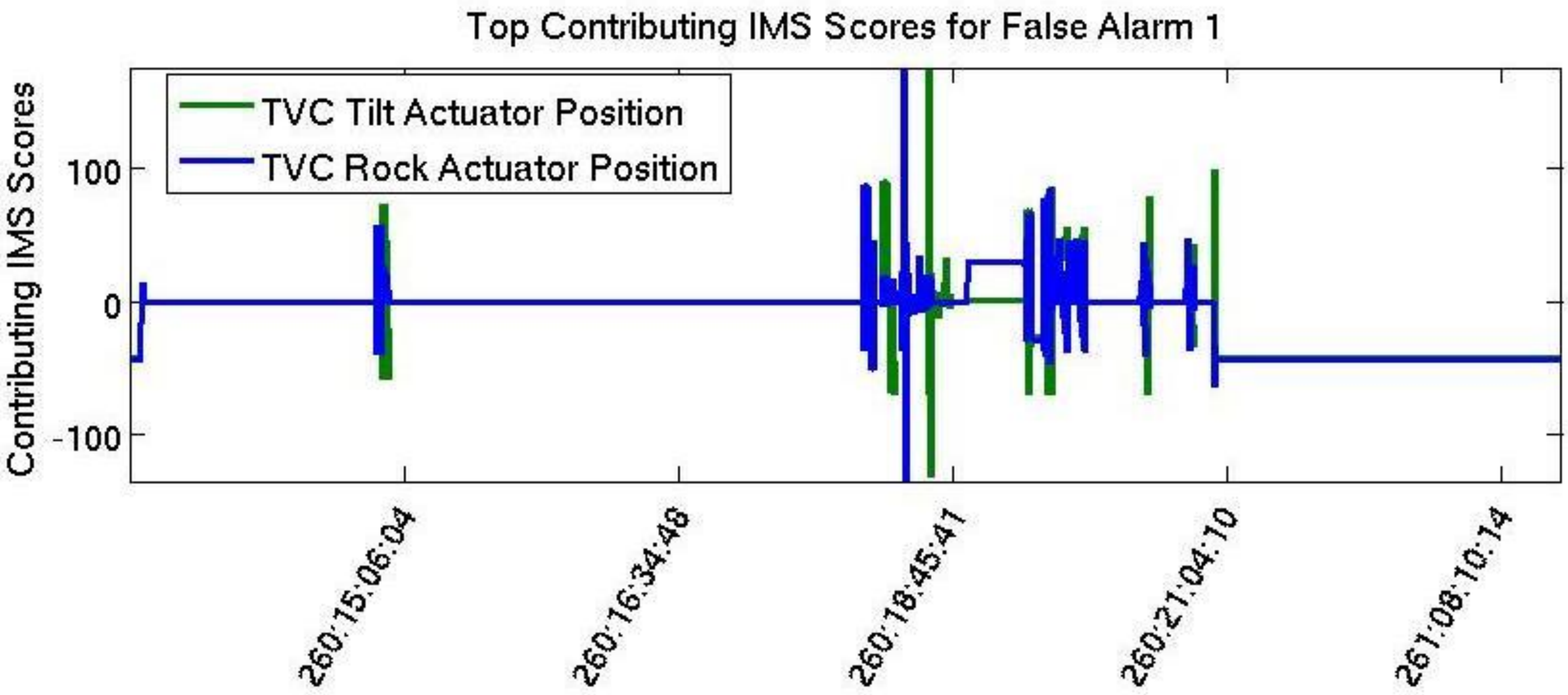
TEAMS/SHINE results on Ares I-X data

- First test on Ares I-X data after initial VAB power-up
  - Some false alarms caused by differences between Shuttle and Ares I-X TVC test procedures
  - SHINE rules fixed
- Shortly before launch, some false alarms caused by data dropouts
  - We had expected dropouts during ascent, but not before launch
  - After launch, we modified our code to detect data dropouts
  - Ran modified code on launch data with no false alarms
- Ares I-X had no failures in the systems we modeled.
  - Prototype had no missed detections and no correct detections.

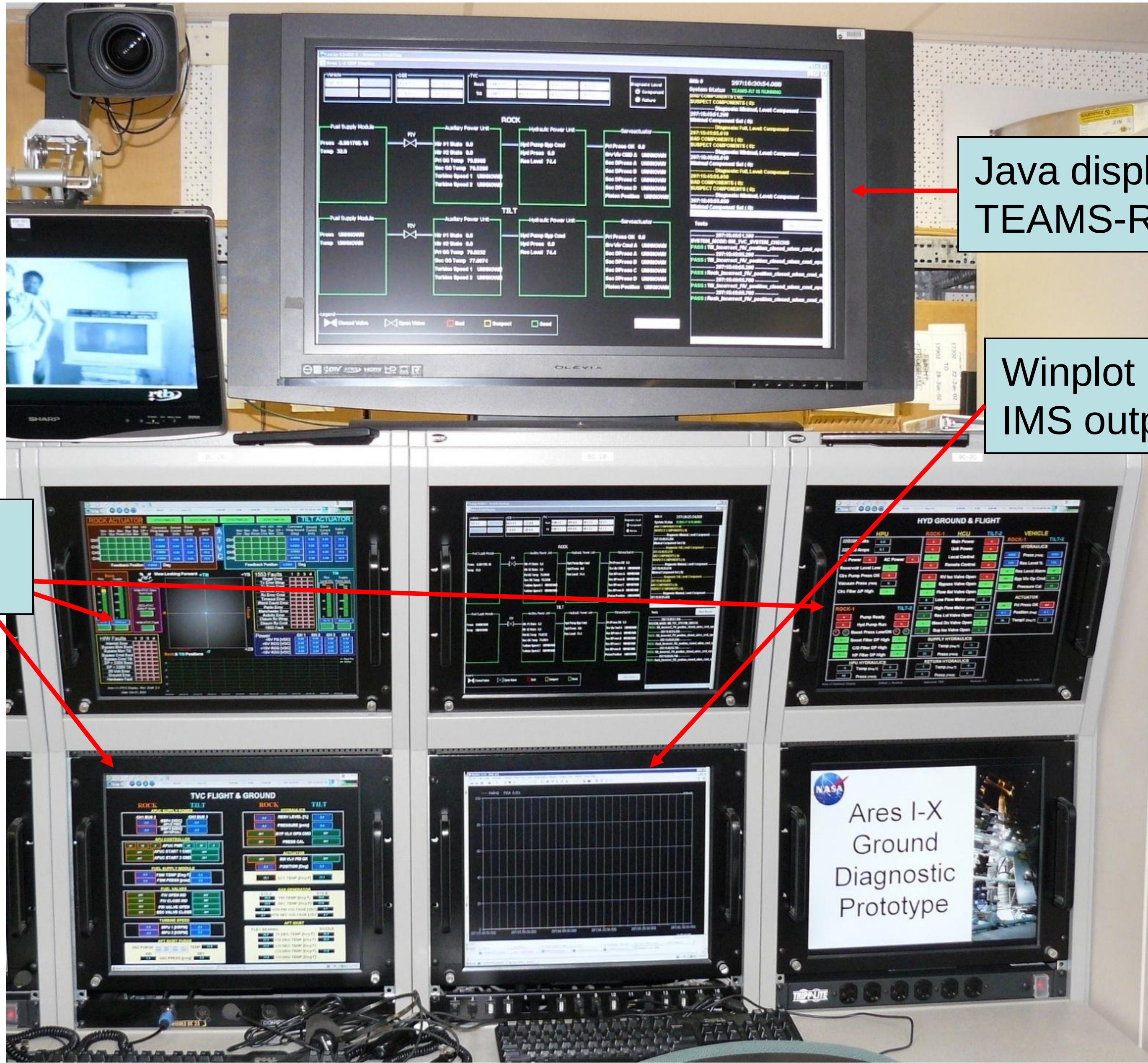
IMS results for Ares I-X VAB data



IMS Contributing scores for Ares I-X VAB False Alarm 1



GDP Console in Hangar AE

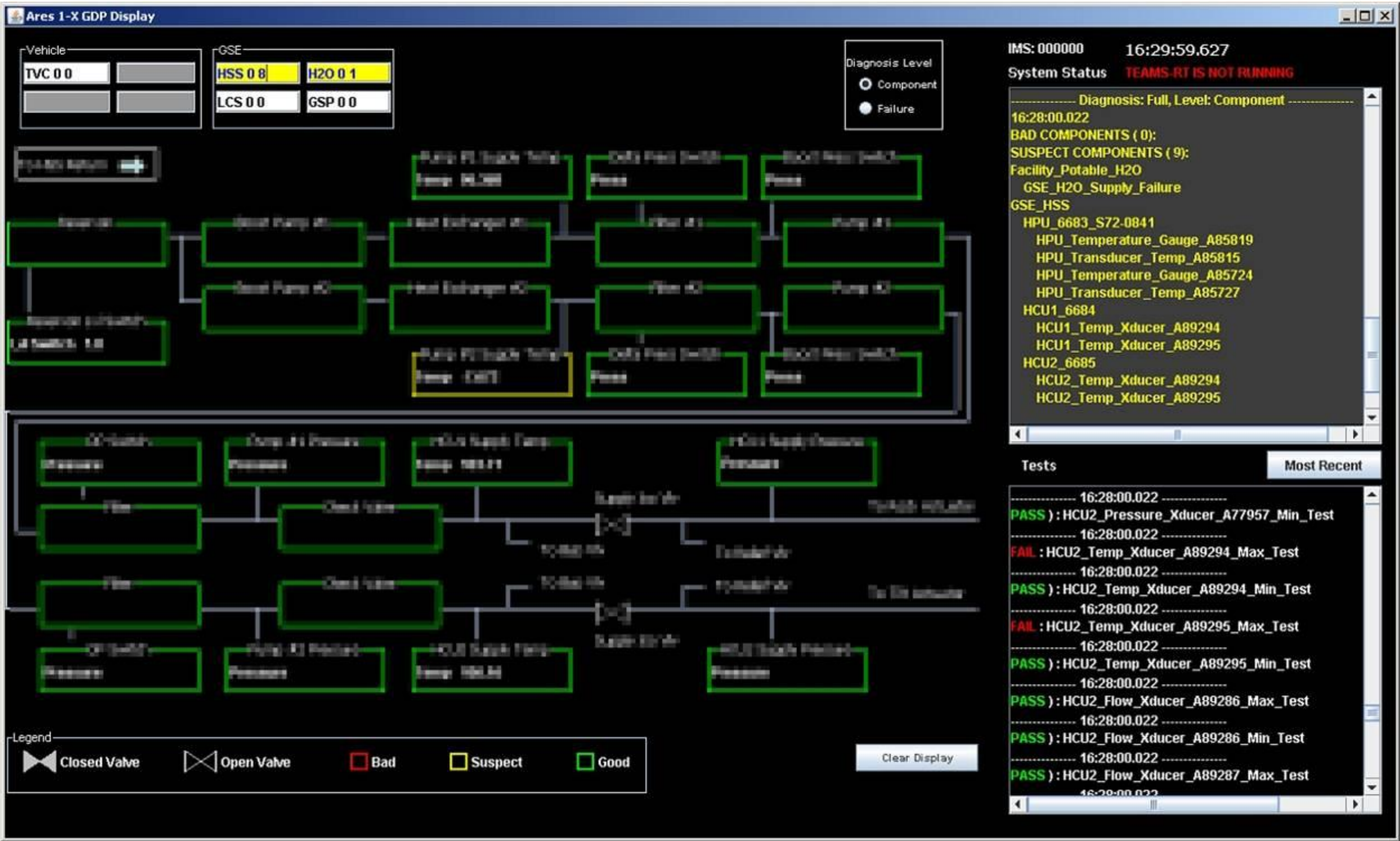


Java display of TEAMS-RT outputs

Winplot display of IMS outputs

Ares I-X Iris displays

GDP Screen Shot



Computational Performance

- Dell Precision M4400 laptop with an Intel Core 2 Quad Q9300 CPU running at 2.53 GHz and 4 GB of DRAM
- 655 components, 893 failure modes, 263 tests, 281 measurements

Process	CPU	DRAM
TEAMS (includes TEAMS-RT, the SHINE rules, the C test logic, and the data interface code)	8%	12 MB
IMS (including its data interface code)	1%	11 MB
Java display (including JVM)	1%	29 MB
Data playback software	18%	56 MB
Plotting tool	1%	12 MB
Windows XP Operating System	5%	410 MB
Total	34%	530 MB

Lesson Learned: Appropriate Roles for Model-Based and Data-Driven Tools

- Model-based tools should be used to detect failure modes that are well understood.
- Data-driven tools should be used to detect unknown failure modes.

Conclusions

- Automated pre-launch diagnostics can help increase safety, reduce cost, and reduce launch delays.
- The Ares I-X Ground Diagnostic Prototype helped to demonstrate and mature automated fault detection and diagnostic software that can be used in future missions.
- GDP demonstrated the feasibility of integrating 3 methods and of integrating the vehicle with the ground systems.
- IMS had some false alarms, as expected, since not all anomalies are failures.
- We believe that the number of false alarms in IMS will decrease over time as more data becomes available.
- We believe the TEAMS false alarms could have been avoided with better V&V.